FDS Pokémon Battle Prediction 2025: Team Eeveelution Report

Michelangelo Crea (1993024) Lorenzo Musso (2049518) Giulia Pietrangeli (2057291)

Abstract

This report details our system for the FDS Pokémon Battles prediction 2025 competition. The architecture is an ensemble of three gradient boosting models: CatBoost, LightGBM, and XGBoost. A key aspect was developing unique, model-specific feature engineering pipelines to maximize ensemble diversity. We explored two primary ensemble methods: a "Level 1" stacking approach (producing two submissions from a Logistic Regression model and a "best model" selection process) and a simple blending ensemble.

1. Feature Engineering & Base Models

Our approach relied on creating three independent "Level 0" models, with diversity driven primarily by distinct feature engineering (FE) pipelines.

1.1. CatBoost Pipeline

This pipeline executed a multi-phase FE process. Key feature categories included:

- **Dynamic Battle Aggregates:** Tracking HP, faints, and stat boosts over time.
- **Static Matchups:** Comparing the lead Pokémon's types against the opponent's lead.
- **Move Characteristics:** Analysis of STAB, healing, and status-inflicting move properties.

All categorical features were processed using One-Hot Encoding.

1.2. LightGBM Pipeline

A second, distinct feature set was engineered for LightGBM. Notable features included:

- **Battle Momentum:** Calculating the slope and volatility of the cumulative HP advantage over time.
- **Information Advantage:** Tracking the number of Pokémon "revealed" (seen) by each player.
- **Timeline Summaries:** Aggregate features such as damage dealt in the first 2 turns.

1.3. XGBoost Pipeline

The XGBoost pipeline created the same feature set as LightGBM. Other important features that we created are:

- **Team Role Analysis:** Identifying the "fastest sweeper" or "bulkiest wall" on the team.
- **Defensive Cohesion:** Calculating the maximum number of team Pokémon weak to a single attack type.

2. Final Ensemble Strategies

We compared methods to combine the predictions of the three base models.

2.1. Method 1: Stacking Meta-Models

We implemented a "Level 1" stacking ensemble by training meta-models on the cross-validated (OOF) predictions from the base models. For each base model, OOF predictions were generated using a 10-fold StratifiedKFold. These were then saved as `.npy` files and stacked to form the final "meta-training" dataset. This process generated two separate submissions:

1a. Logistic Regression Model A dedicated **Logistic Regression** model was trained on the OOF predictions. This model learns optimal linear weights for combining the base models and was used to create one submission file.

1b. Best Model Selection Separately, we conducted a comparison to find the best meta-model. We trained and evaluated several models on the OOF data: **Logistic Regression**, **RidgeCV**, a **LightGBM (L1) model**, a **'Hard' Voting Classifier**, and a **'Soft' Voting Classifier**. The model with the highest OOF accuracy was saved as the 'BEST' model for a separate submission.

2.2. Method 2: Simple Blending

As an alternative, we implemented a simple blending ensemble. This method loaded the final probability `.csv` files from all three models and calculated an equal-weighted (1/3) average for the final prediction.

2.3. Final Submissions

Our three final submissions reflect these distinct strategies:

1. `submission_stacking_logreg_3models.csv`
2. `submission_stacking_BEST_L1_model.csv`
3. `submission_blended_LGBM_CAT_XGB_CLASS.csv`